L18: CLOUD

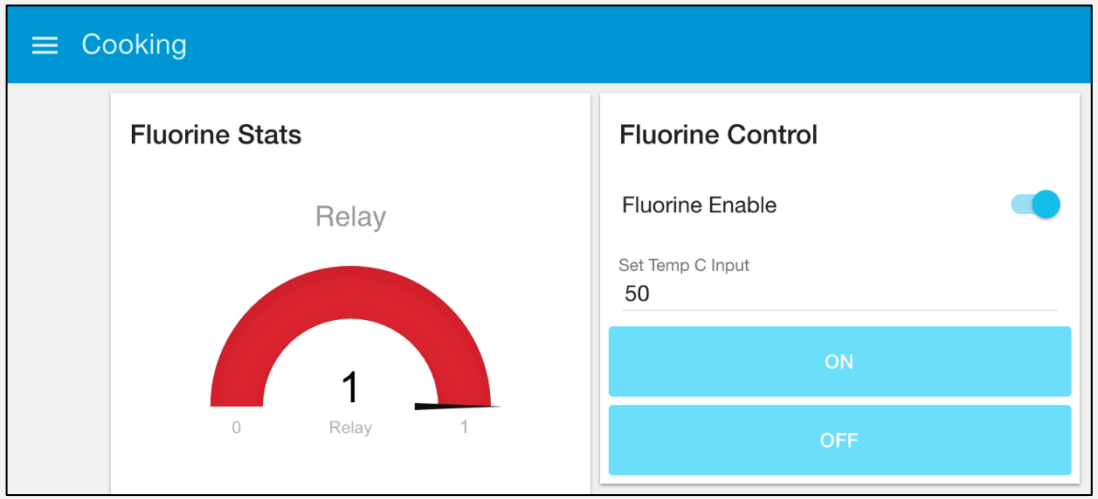
ese516: IoT Edge Computing

Wednesday April 3rd 2019

Eduardo Garcia - edgarc@seas.upenn.edu

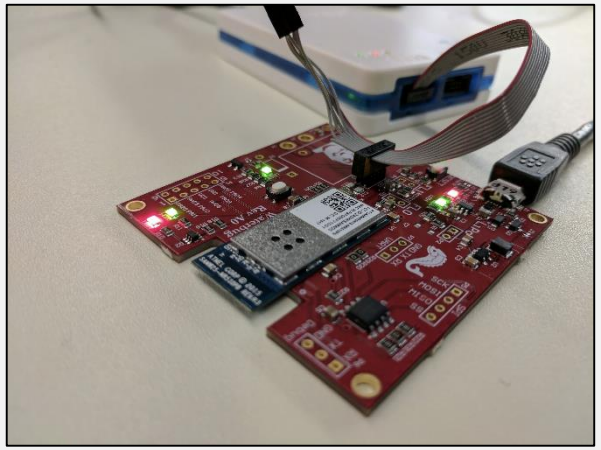
CLOUDY WITH A CHANCE OF IOT

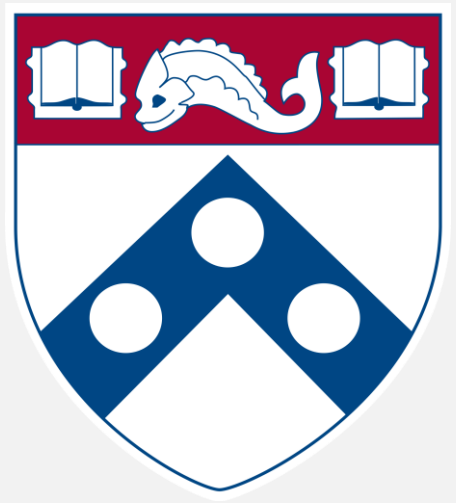
DATA COMMUNICATION LAYER



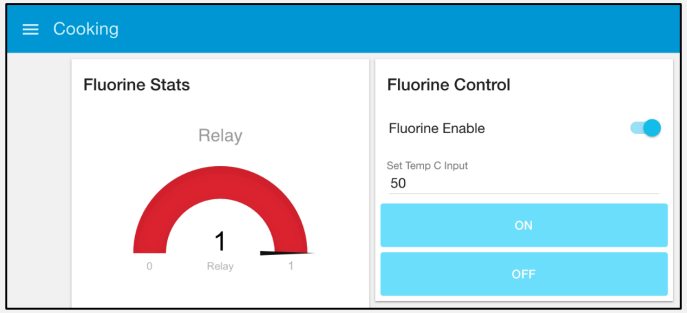
BACKEND & DISPLAY


IoT Device
MQTT Broker
Node-Red





IBM **Cloud**


Cooking
Fluorine Stats
Relay
1
0
Relay
1
Fluorine Control
Fluorine Enable
Set Temp C Input
50
ON
OFF

IOT DATA PROTOCOLS

IOT DATA PROTOCOLS

- MQTT – Message Queueing Telemetry Transport
  - Many to many approach – broker based system
  - Long lived TCP connection
- CoAP – Constrained Application Protocol
  - Single to single – client server approach
  - Send and receive UDP packets

https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php

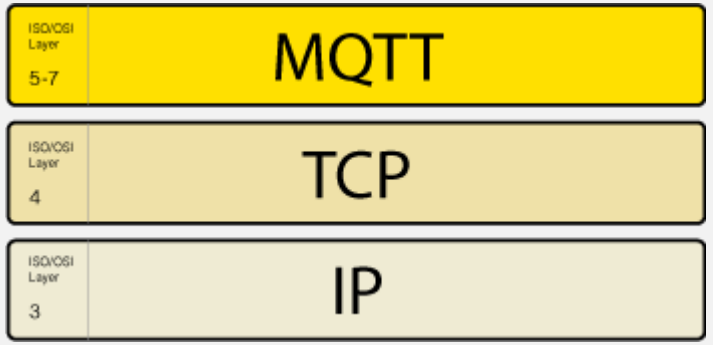
WHY NOT HTTP?

- HTTP has a connection per data transfer
  - One for writing, one for reading
  - Not as lightweight
  - Device has to poll server for data

MQTT

WHY MQTT?

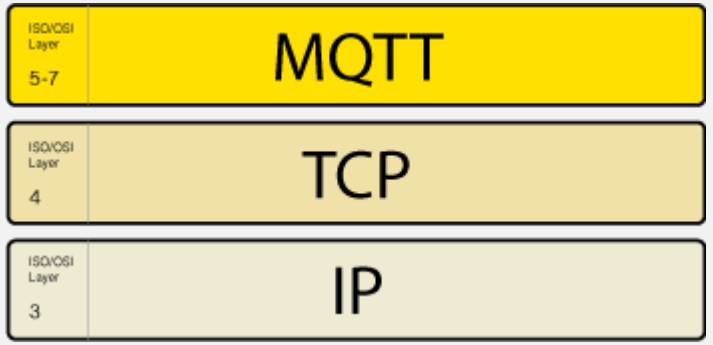
- Message Queuing Telemetry Transport??
- Light-weight code footprint & messages
- Ideal for low bandwidth networks!
- Some assurance of packet reception – quality of service
- Many to many devices – decoupled by using a broker.
  - Don't have to know anything about the devices sending or receiving the packets



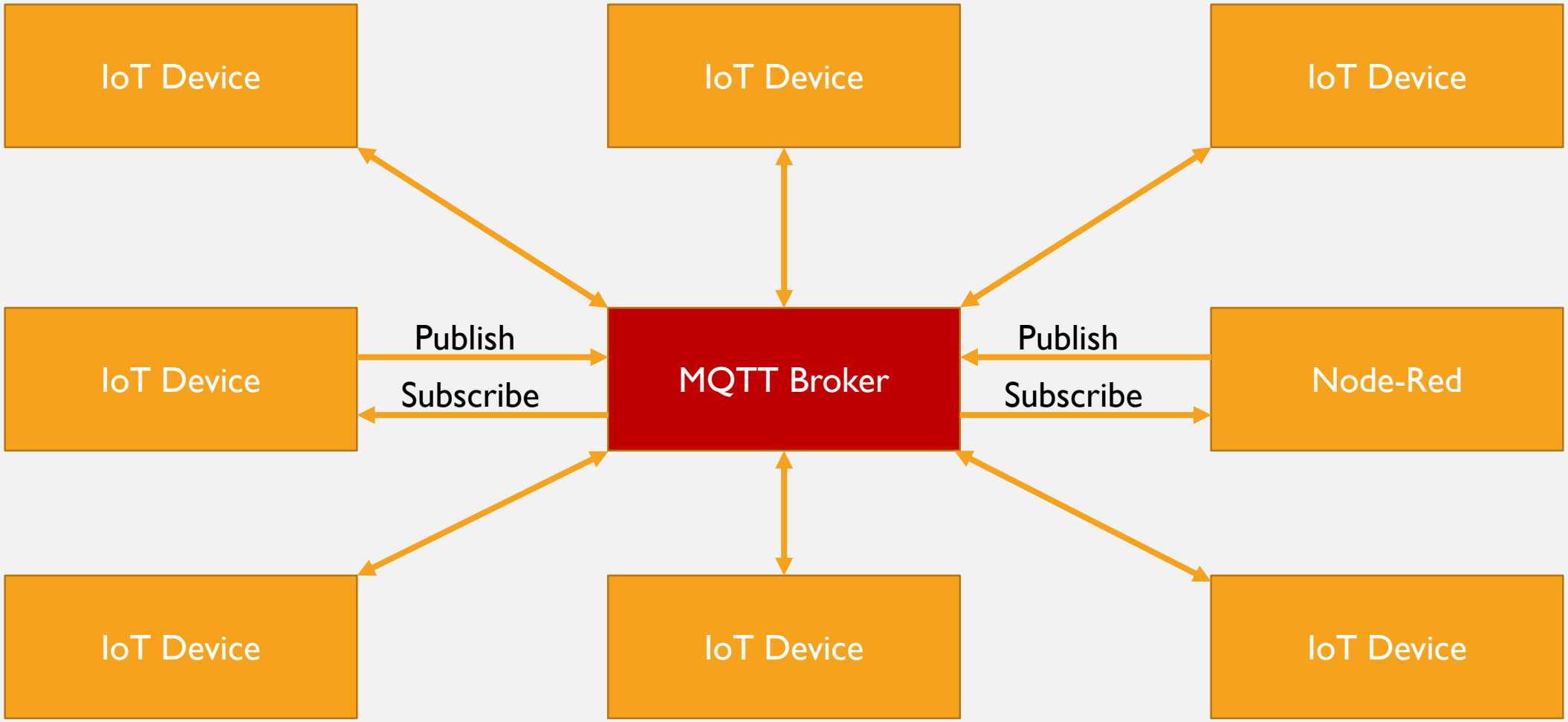
WHY MQTT?

- *MQTT was invented by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom, now Cirrus Link) back in 1999, when their use case was to create a protocol for minimal battery loss and minimal bandwidth connecting oil pipelines over satellite connection. They specified the following goals, which the future protocol should have:*
  - *Simple to implement*
  - *Provide a Quality of Service Data Delivery*
  - *Lightweight and Bandwidth Efficient*
  - *Data Agnostic*
  - *Continuous Session Awareness*

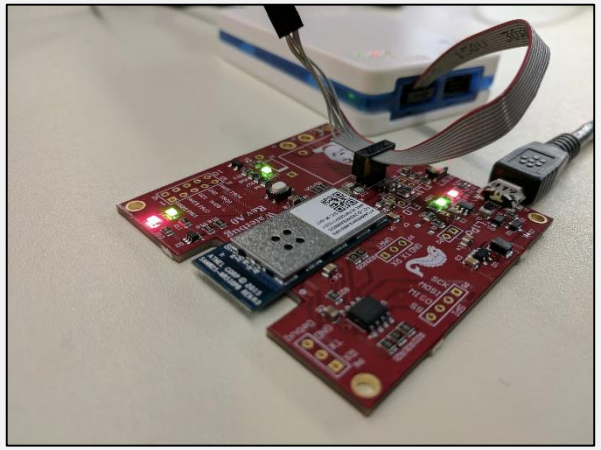
https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt

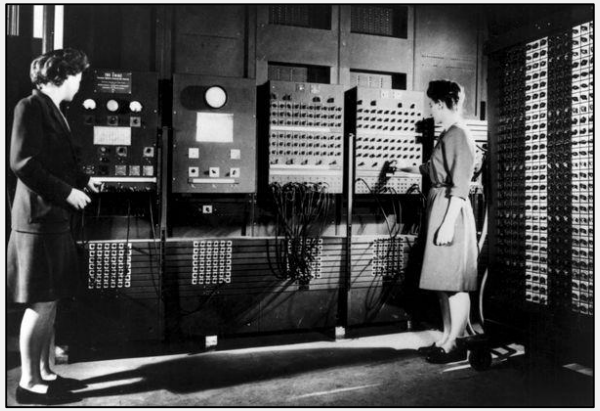



IoT Device
Publish
Subscribe
MQTT Broker
Publish
Subscribe
Node-Red

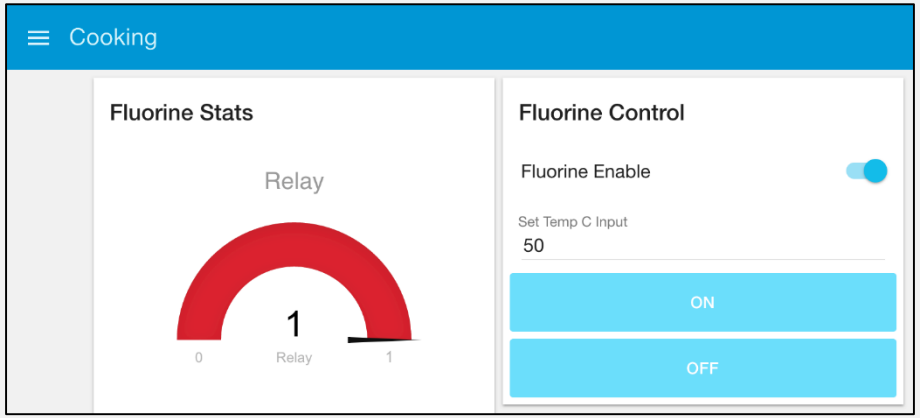

IoT Device
IoT Device
IoT Device
IoT Device
Publish
Subscribe
MQTT Broker
Publish
Subscribe
Node-Red
IoT Device
IoT Device
IoT Device


```
...
case MQTT_CALLBACK_CONNECTED:
{
        mqtt_subscribe(module_inst, MAIN_CHAT_TOPIC "#", 0);
```





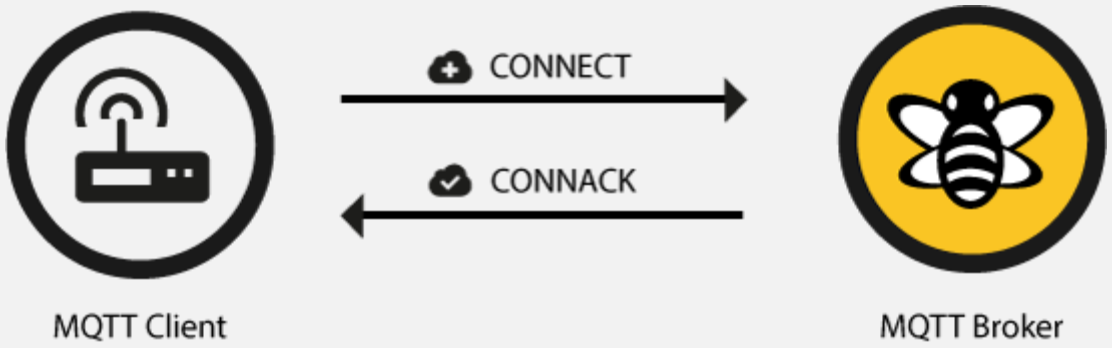
ENIAC
deet.seas.upenn.edu



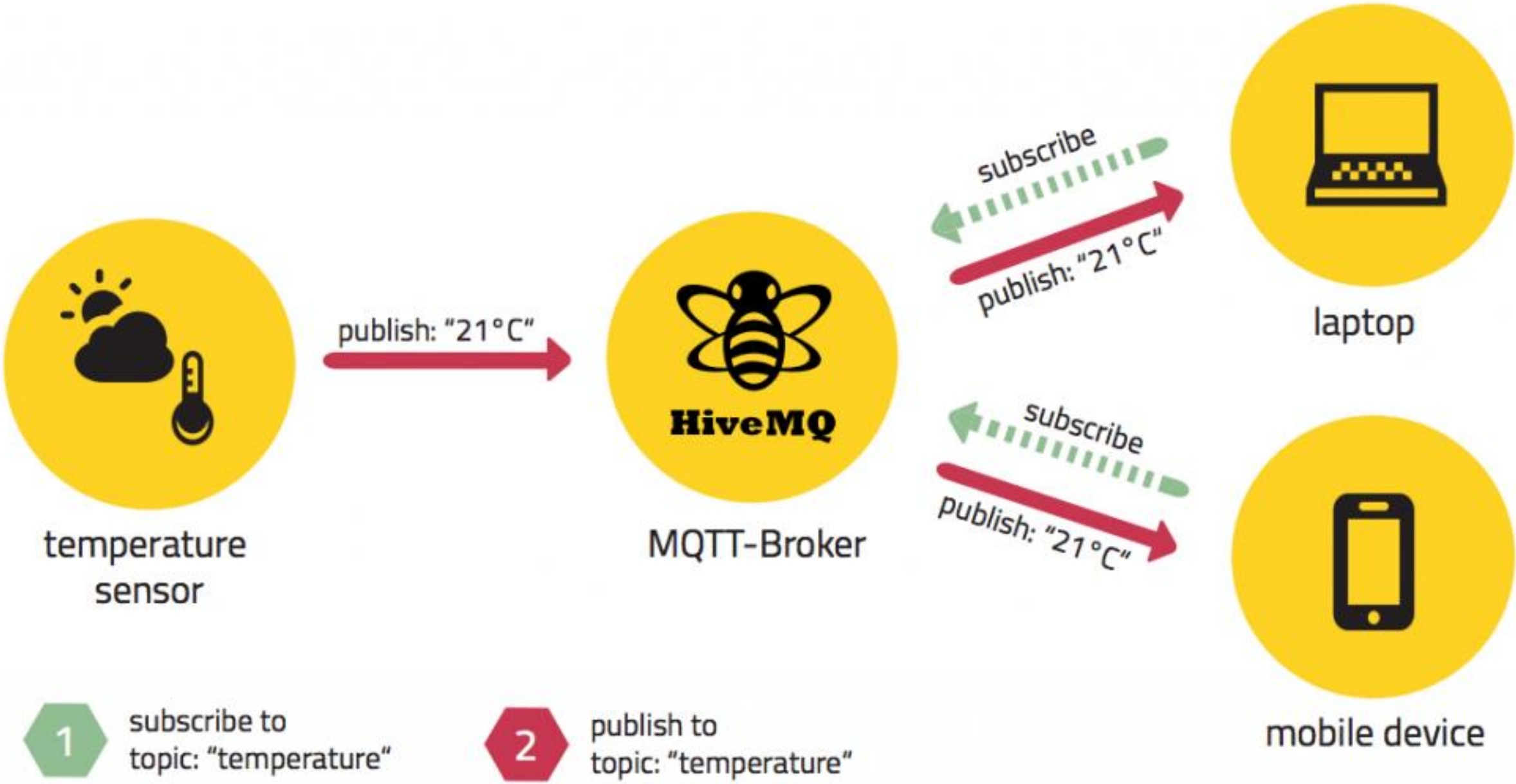
IBM Cloud

MQTT STRUCTURE

- **Topics:** Like a sub-reddit, subscribe & publish through the broker
- **Subscribe:** write to a topic
- **Publish:** listen to a topic
- **QoS:** how the message is delivered
  - 0: At most once
  - 1: At least once
  - 2: Exactly once (what I use)
- **Retain bits:** Set it with your publish packet if you want the value to stay.
  - Devices will receive this value when subscribing.



- Devices connect to the broker.
  - ~80 bytes to connect to server
  - ~20 bytes for publish or subscribe to a topic
- The broker sends back an ACK (acknowledgement) to the device.
- If a message has been retained on a certain topic, it is sent to the device at this point.



https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe

MQTT WILDCARDS

- **Single Level '+'**

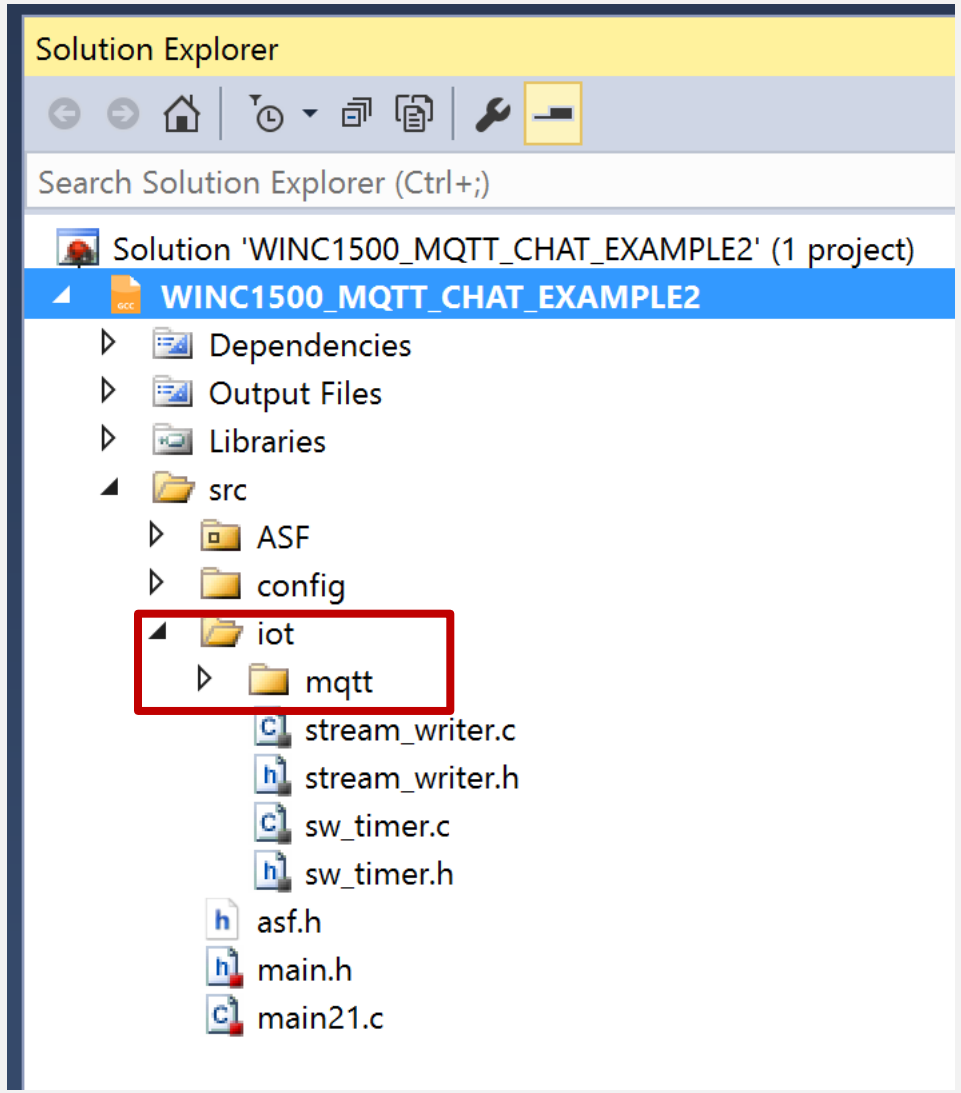
- myhome / groundfloor / livingroom / temperature
- myhome / groundfloor / kitchen / temperature
- myhome / groundfloor / kitchen / brightness
- myhome / firstfloor / kitchen / temperature
- myhome / groundfloor / kitchen / fridge / temperature

- **Multi Level '#'**

- myhome / groundfloor / livingroom / temperature
- myhome / groundfloor / kitchen / temperature
- myhome / groundfloor / kitchen / brightness
- myhome / firstfloor / kitchen / temperature

SAMPLE APPLICATION

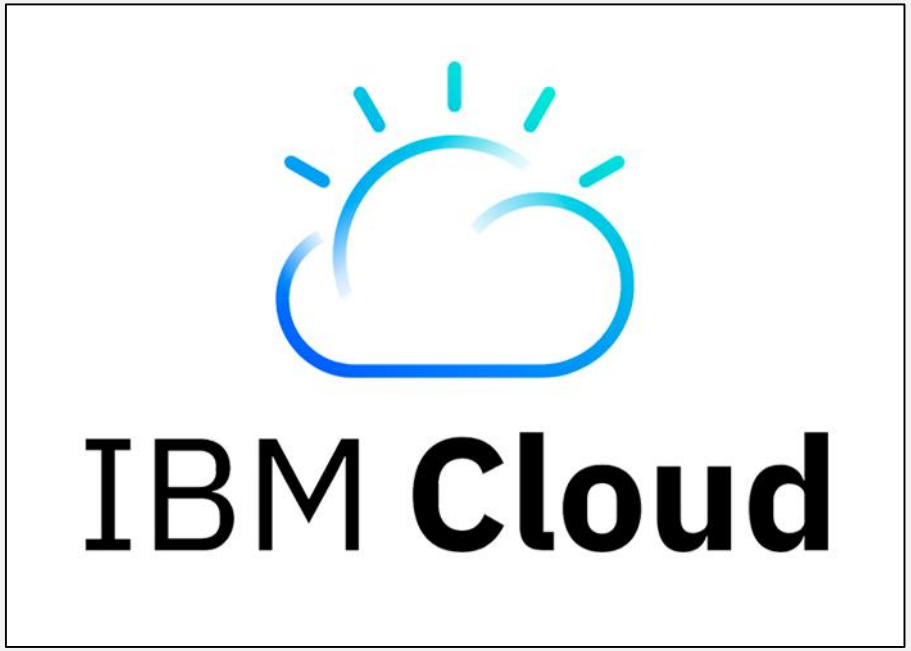
- WINC1500 MQTT Chat Example is the best starting point.
- The **iot** folder contains the critical supporting source code.



IBM CLOUD (BLUEMIX)

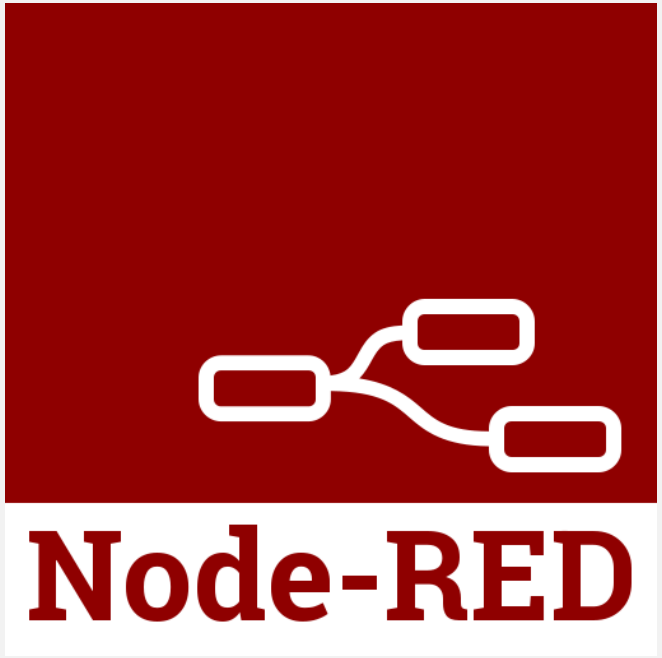
WHAT IS IBM CLOUD?

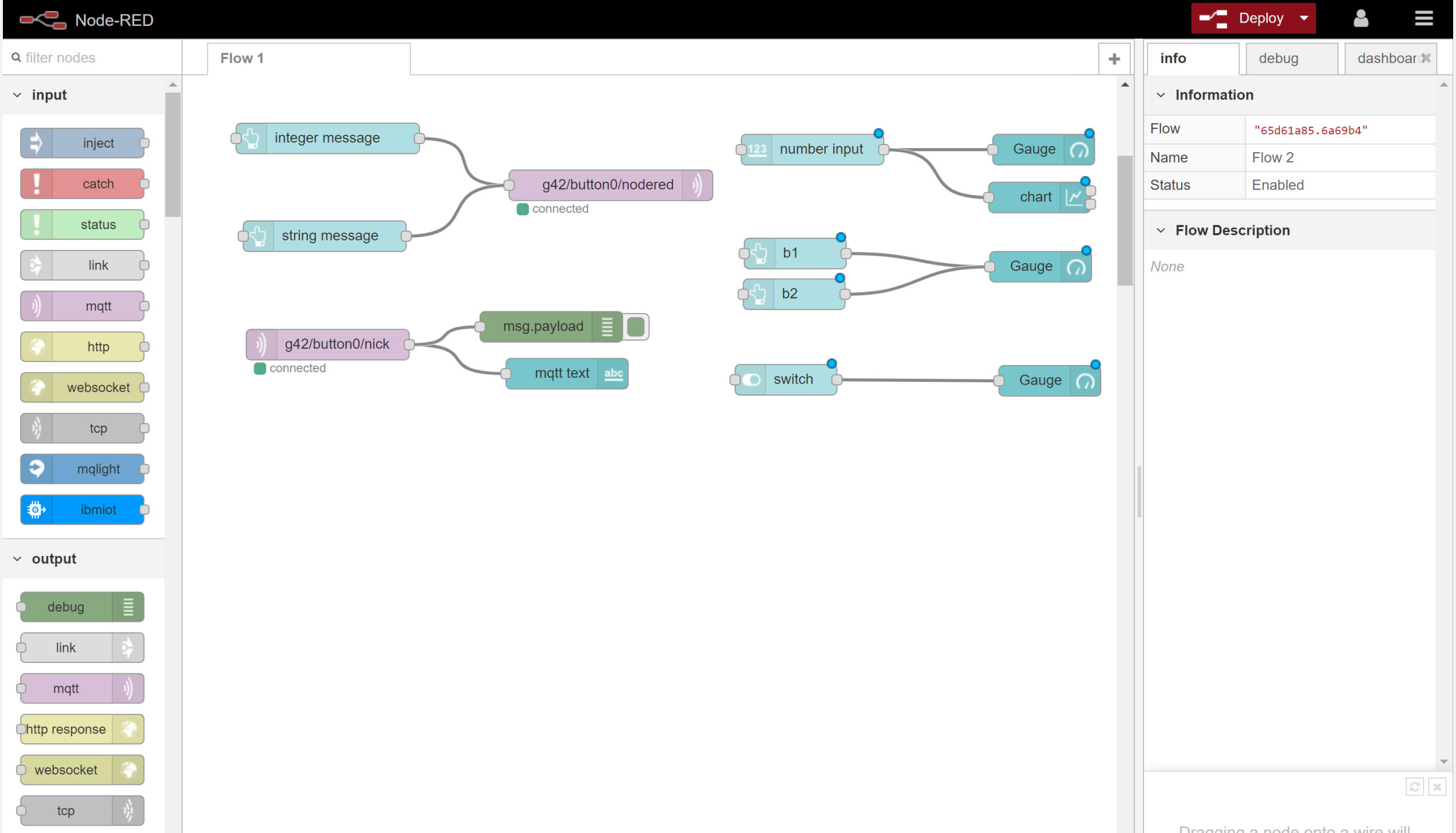
- Front end for displaying data
- Back end for storing data
- Databases for storing data
- Analytics modules for deriving trends
- MQTT broker for handling messages



WHAT IS NODE RED?

- Browser based flow editor
- Open source tool for combining hardware interaction with APIs and existing libraries
- It's an easy way to provide a nice UI (User Interface) without spending a ton of time on it.
- Based on Node.js, so you can make any Javascript blocks you'd like.
- Twitter, Email, Twilio (texting) integrations – sky is the limit
  - You could have people tweet at your Twitter handle to water the plant.
  - Get an email every time your ping sensor triggers.



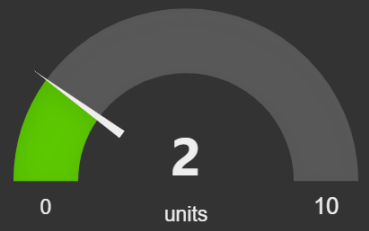

Node-RED
Deploy
filter nodes
Flow 1
input
inject
catch
status
link
mqtt
http
websocket
tcp
mqlight
ibmiot
output
debug
link
mqtt
http response
websocket
tcp
integer message
string message
g42/button0/nodered
connected
g42/button0/nick
connected
msg.payload
mqtt text
abc
number input
Gauge
chart
b1
b2
Gauge
switch
Gauge
info
debug
dashboar
Information
Flow
"65d61a85.6a69b4"
Name
Flow 2
Status
Enabled
Flow Description
None

home

tab1

chart

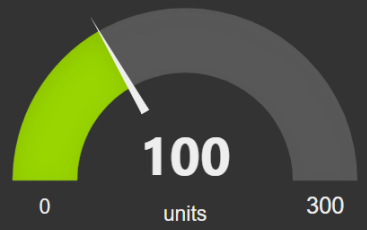
Gauge



number input 2

tab2

Gauge

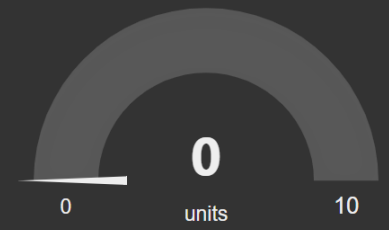


B1

B2

tab3

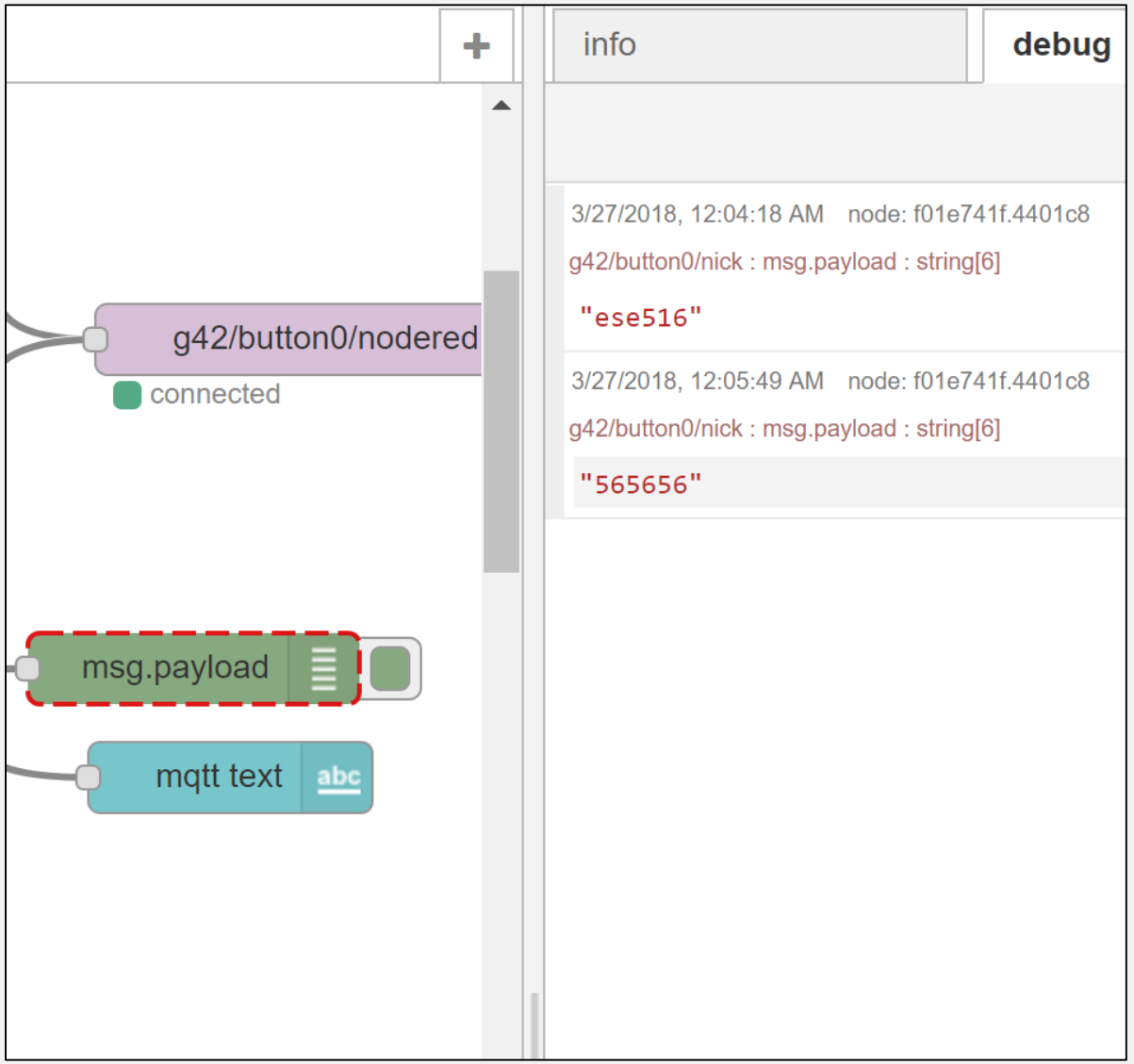
Gauge



switch

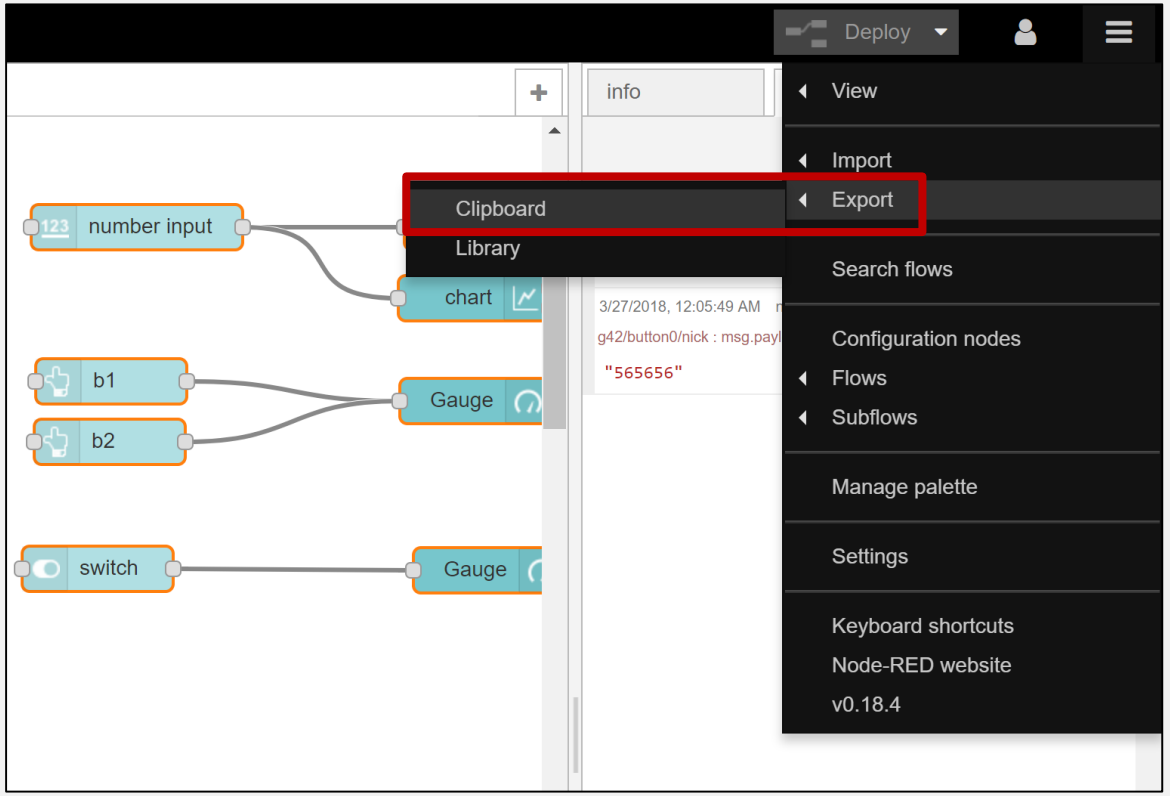
DEBUG BLOCKS

- You can use the debug blocks to print out messages or their attributes.
- Helps if you're trying to parse out your logic!



EXPORT NODE RED CODE

- Select at least one node in your flow constructor zone
- Through the menu, go to **Export > Clipboard**
  - You can then select **All** to export
- Now, you can keep a local / revision controlled copy of your Node-Red environment



GET YOUR FREE PROMOCODE

- Instructions in **Google Drive > Resources > IBM Cloud / Bluemix Setup**
- https://docs.google.com/document/d/1NIOKgPgqifxtXj1iNvf79K6vyIuhNDQcGyMobX_ksxs/edit